FAWRMT: WITH SPECIAL EMPHASIS ON GRAMMAR DESIGNS AND PARTITIONED PARSING

Andy Wong Man Hon & Suen Caesar Lun
City Polytechnic of Hong Kong

ABSTRACT

This paper describes the prototypical MT system, FAWRMT: Fully-automated Weather Reporting Machine Translation. The system is not developed just to replicate the Canadian system of TAUM-METEO. Based on the consensus that FAMT is feasible in a restricted domain such as weather reporting, this project also aims at experimenting with corpus-based statistics and analysis, variated grammar designs and partitioned parsing to enhance the efficiency of the system.

1. INTRODUCTION

This project aims at developing an English-to-Chinese/ Cantonese machine translation prototype for Hong Kong weather reporting. The system design is corpus-based and special emphasis has been devoted to the grammar designs and partitioned parsing to cope with the complex text structure in this particular sublanguage. Corpus statistics has been the major focus.

1.1 THE CORPUS

We focused on 31 pieces of weather reports collected from the Hong Kong Royal Observatory in July 1992. Each sample contains 4 sections: (1) General Situation -- condition of the current day, (2) Weather Forecast for Hong Kong -- condition of the following day, (3) Outlook -- condition of the next few days, and (4) Forecast for Macau Today -- current condition in Macau. (Note: A sample report is given in Appendix One)

Below is a statistical summary of the corpus:

Total number of words	=	3,940
Total number of different words	=	493
Average number of words per sample	=	130
Average number of sentences per sample	=	12
Average sentence length (complete sentences)	=	21 wds
Average sentence length (incomplete sentences)	=	8 wds
Average sentence length (overall)	=	14.5 wds

2. SYSTEM OVERVIEW

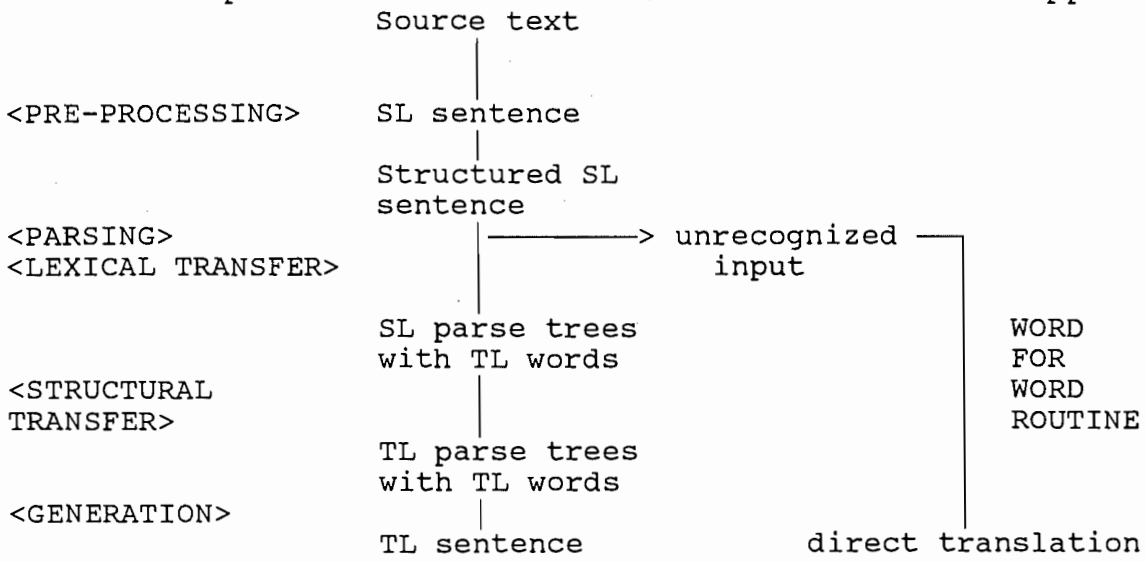
2.1 Basic Characteristics

The system translates single sentences and texts in batch mode with no human intervention. It requires no special formatting on the input apart from the addition of 4 markup symbols to indicate section boundaries, and post-editing is totally eliminated. The program is written in LPA Prolog V2.5 in Eten3/HAN environment, running on IBM pc and accommodating up to 130 words (1 report) per 20 seconds. Up to present it has successfully translated 8 pieces of reports.

To make the system more user-friendly, 2 special facilities are incorporated: (1) double TL Choices: With separate lexicons for Standard Modern Mandarin and Standard Colloquial Cantonese, it allows users to pick the appropriate TL for his own mode of presentation (written or oral). (2) Word-For-Word Routine: The system makes no imprudent guess on unrecognized input elements (wrong spellings / words not found in the lexicons) but triggers this routine to produce a direct translation which suggests what the actual translation should be like.

2.2. Translation Model

We adopt a modified version of the Transfer approach:



(A) PRE-PROCESSING

By identifying typographical features, the system extracts individual sentences from the text, tokenizing each into a machine-readable list form.

(B) ANALYSIS

The parser identifies the structural constituents of the sentence according to the Analysis Grammar, looks up the words in the lexicon, and returns the TL terms and syntactic/semantic information. Finally a SL parse tree is built. The process is syntactically-based, but aided by semantic filtering routines:

LINGUISTIC PRINCIPLES	FUNCTIONS
1. subcategorizations	specifying collocation patterns of verbs and complements in the grammar rules (particularly helpful in handling PP-attachment) e.g. VP --> V_group Complement Complement(iv) --> [] Complement(tv) --> NP

	Complement(dav) --> NP NP (*NOTE: dav - ditransitive verb)
2. selectional restrictions	assigning semantic features to specify what nouns can follow what verbs and can be modified by what adjectives
3. general collocations	specifying whether a verb can be preceded by an auxiliary verb and if so, in what form
4. other syntactic/semantic features	specifying (1) which TL should be selected among homographs, (2) whether an ADJ should be followed by the TL word "de", (3) which classifier accompanies each noun, (4) TL order of adverbs in a compound adverb group

The process runs on a top-down, depth-first and sentence-by-sentence basis (no risk of neglecting contextual factors since intersentencial connection is insignificant in the sublanguage). Morphological treatment is disregarded because of the lack of inflectional variants, while semantic analysis is reduced to applications of semantic features and filtering routines. Since the analysis lexicon and the transfer lexicon are combined into a single bilingual dictionary, Lexical Transfer proceeds along with Lexical Analysis to avoid checking the same lexical entry twice.

(C) TRANSFER

The SL tree is transformed into a TL tree in a top-down, depth-first manner applying transfer rules constructed out of the TL grammar. Lexical routines (short programs) are implemented for special transformations: (1) selection of TL for determiners according to the definiteness of NPs, (2) selection of nominal classifiers, (3) insertion of the TL words "you3" and "de" before nouns and after adjectives wherever appropriate, (4) ordering of

adverbs in TL. Finally a TL parse tree is formed.

(D) GENERATION

In the absence of morphological treatment, this module is simplified and mainly involves the decoding of TL trees to extract and arrange words into linear TL sequences.

(E) DIRECT TRANSLATION

If the SL string contains unidentifiable words or phrases, the system will trigger a lexicon lookup routine to produce a direct translation as reference. This prevents over-translation and guarantees the accuracy of output.

3. GRAMMAR TYPES

Two ideas are involved in the grammar design: (1) Multi-Path Grammar, which supports partitioned parsing, and (2) Statistically-Based Grammar, which derives maximum benefit from corpus analysis to facilitate parsing.

3.1 Multi-Path Grammar

A weather report is a mixture of complete sentences, incomplete sentences and domain-specific phrases. A partitioned parser with multiple "grammar paths" is thus used, with a Phrase Structure Grammar Path for parsing complete sentences, a Semantic Grammar Path for parsing incomplete sentences and phrases, and a Heading Path for parsing section headings. The Semantic Grammar comprises the ATM Grammar (atmospheric conditions), TEMP Grammar (temperatures) and WIND Grammar (wind conditions). An automatic recognizer checks the sentence nature (by identifying key word,

phrases and structures to trigger the correct grammar paths.

3.1.1 PS Grammars

The grammar is similar to a common one except for including 3 "semantic phrases" to handle domain-specific patterns:

(A) TIME phrases - at sentence beginning for temporal indication:

TIME --> [at] number [am/pm] ADVP

e.g. At 11 pm last night, tropical depression Dianna was centred about 620 kilometres south of Kagoshima.

(B) SPEED phrases - nominal or verbal PPs describing wind speed:

SPEED --> [at/of] modifier number [per] [hour]

e.g. (in NP): Strong gusts of around 90 kilometres per hour were recorded at the airport yesterday.

e.g. (in VP): The tropical depression is forecast to move west at around 22 kilometres per hour.

(C) LOCATION phrases - verbal PPs indicating locations of pressure bodies (e.g. depression, trough, ridge...); with 2 components, DISTANCE phrase and DIRECTION phrase:

LOCATION --> DISTANCE DIRECTION
DISTANCE --> modifier number [kilometres]
DIRECTION --> direction [of] proper_noun
e.g. Eli was centred about 560 kilometres east of Manila.

3.1.2 Semantic Grammars

The corpus is full of unusual sentence patterns which are totally different from those of ordinary complete sentences, but can be generalized into regular semantic patterns of their own. It is inconvenient and inefficient to treat them with a syntactically-based grammar, but by a "semantic grammar" constructed out of those patterns. "The grammar may have an upper

level of semantic categories to mark the semantic patterns, and a lower PS level" (Hutchins 229) to indicate the syntactic constituents in the sentence". This is the case in TAUM-METEO (King 264). The idea is also applied in our system, but has been elaborated, modified and refined to cope with the more complicated sentence patterns. The grammar has 3 parts:

(A) ATM Grammar

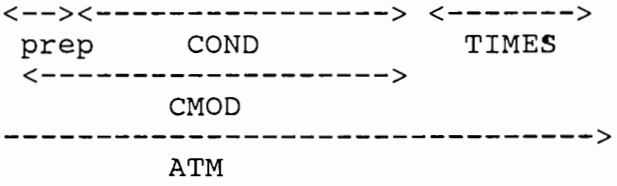
These expressions are special combinations of NPs, ADJPs, ADVPS and PPs. The main rule has 4 components:

ATM --> COND CMOD TIMES COMPL

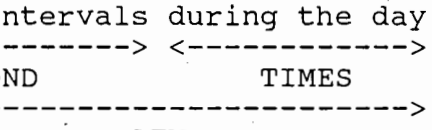
COND	(adj/np/vp)	=	the main atmospheric condition
CMOD	(pp)	=	complementary/accompanying condition
TIMES	(adv/pp)	=	temporal indication
COMPL	(adj/np)	=	complement of the whole ATM phrase

Examples:

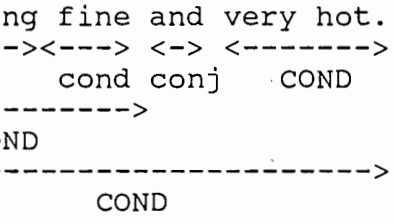
1. Cloudy with scattered showers on Monday.



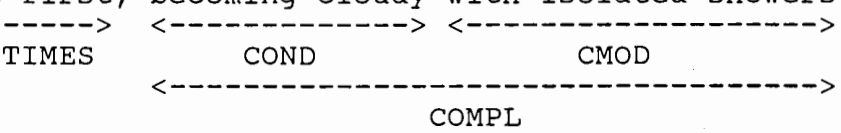
2. Sunny intervals during the day.



3. remaining fine and very hot.



4. fine at first, becoming cloudy with isolated showers.



(B) TEMP Grammar

This is a special case. TEMP expressions are indeed complete sentences. But since they have only 2 patterns, it would be more efficient to treat them as semantic expressions in a "rewriting" manner -- treating the patterns as fixed frames so that the parser will not return the underlying structure of "the minimum temperature is 19 degrees" but will search for "the", "temperature" and "degrees" and pick the correct options from <type> and <verb2>:

[1] Temperatures <verb1> [num] conj [num] degrees.

<verb1> = will range between / will be in the range of/
will range from / are expected to range between

[2] The <type> temperature <verb2> [num] degrees.

<type> = maximum/minimum
<verb2> = will be / will be around / will reach

(C) WIND Grammar

WIND expressions are complex NPs reporting wind conditions:

WIND --> TYPE DIRECTION [winds] COMPL
COMPL --> MODIFIER MAIN PLACE ADV TIME

TYPE (adj)	=	nature of wind condition
DIRECTION (adj)	=	wind direction
COMPL (adj/vp)	=	complement of the whole WIND phrase
MODIFIER (adv)	=	pre-modifier of the complement
MAIN (adj)	=	main condition of the complement
PLACE (pp/adv)	=	locative indication
ADV (adv)	=	post-modifier of the complement
TIME (adv/pp)	=	temporal indication

Examples:

1. Fresh southwesterly winds.
<----> <------------>
TYPE DIRECTION

2. Light to moderate south to southwesterly winds.
<---------------> <---------------------->
TYPE DIRECTION

3. Moderate easterly winds, occasionally fresh offshore later.

```
<------> <------>          <----------> <---> <------> <----->
 TYPE   DIRECTION            MODIFIER    MAIN   PLACE    TIME
                           <-------------------------------->
                                        COMPL
```

4. Fresh to strong southeasterly winds, moderating gradually

```
<---><--><-----> <----------->          <--------> <------->
 TYPE conj TYPE   DIRECTION               MAIN       ADV
<-------------->                       <------------------>
     TYPE                                      COMPL
```

3.2 Statistically-Based Grammar

Grammar clauses are arranged in the program according to their relative frequencies of application. This minimizes back-tracking during run time and enhances processing efficiency.

4. SYSTEM STRUCTURE

4.1. Program Design

The actual program is constructed out of the theoretical translation model. It has the following components:

	<LEVEL 1>	<LEVEL 2>	<LEVEL 3>
		Pre-processor	
SYSTEM -->	Manager Program	PS Translating Program	Main Program Body Module 1: ANALYSIS Module 2: TRANSFER Module 3: GENERATION Chinese Lexicon Cantonese Lexicon
		Semantic Manager Program	(1) ATM Program (2) TEMP Program (3) WIND program

The Manager Program takes overall control of the program by identifying the different sections in the text and triggering grammar paths. Actually, Section 1 contains only complete sentences while the other three contain only incomplete (semantic)

sentences. This signals when to consult the PS program and the Semantic ones.

Still the system has to select among the three Semantic paths. This is done by the Semantic Manager Program, in which an automatic recognizer looks for key words, phrases and structure to identify the sentence nature, with the following algorithm:

INPUT = input sentence

IF INPUT contains the key words "temperature(s)" & "degrees" in the format [...temperatures ... degrees] THEN goto TEMP

ELSE IF INPUT contains the word "winds" at the end or at a clause boundary THEN goto WIND

ELSE IF INPUT starts with an ATM-ADJ, ATM-ADV or ATM-COND word (check the ATM Lexicon) THEN goto ATM

ELSE goto PS

Else goto WORD-FOR-WORD

The key structures are collected from detailed corpus analysis and are reliable at least in handling the 8 pieces of samples.

The Pre-processor formats each sentence into an analyzable form which then goes through 3 translation modules. The process repeats until all sentences in the section have been translated.

The Manager Program continues to search for the other sections, consulting the corresponding grammar path(s) and translating the sentences until all sections have been processed. The process takes about 20 seconds on average.

Input containing unrecognizable elements can still pass the Pre-processor which only checks typographical features in sentence extraction and tokenization. It is not until the input reaches the parser that it is rejected. This is very important

since the unknown elements, though untranslatable, have to be localized into normal tokens so that they can be reported in the the suggested translation produced by the word-for-word routine.

With a pre-processor converting input into machine-readable forms, there need not be any stylistic formatting on the input. Similarly, as the generator decodes all formats and structures in the TL tree, the output text requires no post-editing.

4.2. Lexicon Design

Every translation program has a bilingual lexicon (either the Chinese lexicon or the Cantonese lexicon is consulted once). In the absence of morphological analysis, the lexical entries are simplified. There are no details of agreement, tense, gender, and inflections but only parts of speech, SL and TL terms, and syntactic/semantic information for parsing and transfer.

4.3. Programming Details

4.3.1 Implementation of Rules

Grammar rules and transfer rules are converted to C-Form (Condensed Form) to be processed "deterministically and in real-time" (Krulee 202). A grammar is in C-Form if it is "context free" (Krulee 9) and for each nonterminal, X, there exists at most one rule in the form X --> A,B (A is a terminal, B is anything) and at most one rule X --> e (e is an empty string). For example, the VP rule has multiple patterns which need to be organized in a better way for economy and efficiency:

```
ORIGINAL RULES: vp --> verb np advp
                vp --> verb np pp
                vp --> verb np pp advp ...
                vp --> verb
```

RULES IN C-FORM: vp --> verb M1

M1 --> np M2
M1 --> e
M2 --> advp M3
M2 --> pp M3
M3 --> advp
M3 --> e

4.3.2 Modular Programming

The theoretical model of the system is already modular. In implementation, the design is reserved by keeping different modules apart as "black boxes" with no interference in between beside input and output passages. Even the different components in a module are separated in some cases. This enables localization of errors, integration of new rules, strategies and SL-TL pairs (Picken 91). However, LPA Prolog allows only structural but not functional modularity since predicates within a module are not entirely local and invisible in others.

4.3.3 Processing Efficiency

A reliable way to upgrade program efficiency in Prolog is to reduce stack overheads, with first argument indexing -- using the first argument of a predicate as index to distinguish the clause from others instead of keeping the clauses as separate rules. This reduces unnecessary growth of backtracking stack and promotes deterministic parsing. For example, our PS parser contains the following rules:

```
p(1,ADV,PP,P0,P):- advp(A1,P0,P1),p(4,A2,PP,P1,P).
p(1,ADV,PP,P0,P):- pp(PP,vp,P0,P1),p(2,ADV,P1,P).
p(2,ADV,P0,P):- advp(ADV,P0,P).
p(2,[''],P,P).
p(3,AJP,ADV,PP,P0,P):- adjp(AJP,_,P0,P1),p(4,ADV,PP,P1,P).
p(3,adjp(epsilon),[''],pp(epsilon),P,P).
p(4,ADV,PP,P0,P):- pp(PP,vp,P0,P1),advp(ADV,P1,P).
p(_,[''],pp(epsilon),P,P).
```

5. EVALUATION AND DISCUSSION.

1. Grammar Designs

We have adopted 4 approaches in designing the grammars: Statistically-Based Grammar, Multi-Path Grammar, Semantic Grammar and Rewriting-Based Grammar.

(1) Statistically-Based Grammar

This is to take advantage of Prolog's backtracking mechanism. If clauses are arranged out of their relative frequencies of occurrences, the possibility of backtracking will be minimized. Each grammar rule is given a statistical index according to the result of a concordance analysis, which decides their position in the program.

The idea turned out to be workable. An indexed program made fewer backtrackings on average and ran faster. Though the working space saved is small in a single case, the effect is maximized in larger program segments. In the dictionary module, for example, there are hundreds of entries and the effect is more obvious. This is particularly effective in compound-term entries (as the following PN entries taken from the PS Translating Program) where SL terms are implemented as lists which have to be scanned through each time. Obviously the statistical method saves much effort in the long run.

```
dicts(pn,[south,china],TL,[Features]).
dicts(pn,[hong,kong],TL,[Features]).
dicts(pn,[northern,guangdong],TL,[Features]).
```

Nevertheless, the method is sometimes inapplicable:

(1) Recursions:

```
    ADJP --> adj
    ADJP --> adj  ADJP
```

(2) Clauses of extremely low frequency: <INDEX>

```
    NOUN --> n                    (64.7%)
    NOUN --> pn                   (22.8%)
    NOUN --> n_def                ( 9.5%)
    NOUN --> n_cpd                ( 2.0%)
    NOUN --> pron                 ( 1.0%)
    NOUN --> n_dummy              ( 0.2%)
```

(3) Optional clauses:

```
    OPTREL --> []                 (94.2%)
    OPTREL --> [that]  VP         ( 5.8%)
```

In (1), even if the second clause is used more often, it must not come before the first -- the terminating condition of the recursion. In (2), n_dummy has only one entry but becomes the first clause. The reason is obvious: if the system has to fire this clause, it has to make unnecessary efforts to go through the preceding ones. As we now place n_dummy at the top, even when it is not the one to be chosen, only one backtracking is wasted. In (3), after the empty-list rule is formed into a Prolog clause it becomes an all-pass clause as an "optional" way out in case the main clause does not fit. If it is located at the top, all incoming checks will pass and become meaningless. For these pragmatic reasons, the technique is sometimes inapplicable. But the idea itself is feasible in most cases anyway.

(2) Multi-Path Grammar

It is inefficient to cover the different types of sentences in the corpus with a single parser. Even if such a parser can be produced, it will be clumsy and inefficient. It is also unwise

to implement so large a segment in one program module. We therefore partitioned the grammar into separate paths to handle different types of expressions.

Although more programming space is required to implement the grammar so that other program designs must be economical enough to compensate for the loss, the advantages seem to compensate for all that. It solves the problem of inconsistent sentence patterns. While individual grammar paths are small in size, more evaluation space is spared which ensures faster processing. It also supports modularity through the division of labour.

The value of sublanguage in MT is that the domain-specific patterns bring great convenience in building the grammar. If we use a single PS Grammar and arbitrarily fix all patterns under the traditional PS categories, we will possibly ignore their semantic natures. The parse tree produced will also be unable to reflect the sentence content.

(3) Semantic Grammars

To utilize the domain-specific patterns, we may either reserve the use of PS grammar but accommodate the rules to the patterns, or generalize the semantic natures of patterns into a semantic grammar. In the latter case, a completely new grammar is produced which analyzes a sentence in a slot-filling manner. See how the following ATM sentence is analyzed in both ways:

[Sunny with some showers in the morning, hot in the afternoon.]

```
PS:  SENTENCE --> ADJP PP PP ADJP PP      ADJP --> adj
                                          PP --> prep NP
                                          NP --> det n
```

Semantic: SENTENCE --> COND CMOD TIME COMPL

COND --> sunny
CMOD --> with some showers
TIME --> in the morning
COMPL --> hot in the afternoon

Obviously the modified PS grammar is clumsy and hard to comprehend, nor does it accurately reflect the functional and structural relationships between the constituents. If the pattern occurs regularly in the corpus and has its own semantic structure, why not generalize it into a semantic grammar? Actually, the semantic grammar gives a more reasonable representation of the constituents in a general way, representing a universal framework. Sentences fit into it with their subparts filling in the appropriate slots. While the grammar is specifically designed and corpus-based, it has no unnecessary element. It also does not stick to any traditional framework and can be flexibly reformed and extended to cope with problems in parsing so as to resolve structural ambiguities. More importantly, the grammar, enclosing a patterning framework of the sublanguage, becomes a special type of "knowledge representation based on semantic primitives" (Nirenburg 31), and is surely an economical one since the grammar and the knowledge schema are combined into one!

At first sight the idea runs the risks of lacking generality and poor syntactic indication since the semantic types may have no straight forward correspondence with the syntactic structures. It also takes extra time to learn. But while the grammar is specific to a subworld, the rules need not be so general. Fur-

thermore, the grammar can include syntactic categories at a lower level to represent the syntactic nature (Hutchins 229).

(4) Rewriting-Based Grammar

The TEMP program is the shortest and runs fastest among the four, implying that this grammar saves programming space and enhances processing speed. The only disadvantage remains the inability to reveal the syntactic structure of the sentences. But while the sentences covered by the grammar appear regularly, their structures should be familiar. Is there any need to check the structures over and over again?

5.2. Problems and Solutions

During the system development, various problems have arisen, and have been tackled completely or partially in different ways.

(1) Ambiguities

Owing to the restricted vocabulary in the sublanguage, lexical ambiguity is rare and arises only when there are several TLs for the same word -- translational ambiguity (King 262). Three solutions have been adopted:

(A) SELECTIONAL RESTRICTIONS:

e.g. "rain" has different TLs in the following contexts:

1. Rain became much less frequent and intense yesterday.
2. More than 70 millimetres of rain fell over the territory.

WORD	TL	SEMANTIC FEATURE
rain[1]	yu3 shi4	[RAIN]
rain[2]	yu3 liang4	[QUANT]

QUANT phrases must take a [QUANT] noun, so rain[2] is chosen in sentence 2. The mismatch between [QUANT] and the context of

Sentence 1 forces the selection of the [RAIN] entry.

(B) FUNCTIONAL CATEGORIZATIONS: Some word classes are highly categorized according to their structural roles so that correct TLs can be chosen in different contexts:

conjunctions	-->	pre-sentence conj, inter-sentence conj, intra-np/vp conj, inter-vp conj
nouns	-->	n, proper n, compound n, definite n
verbs	-->	v, phrasal verb

(C) FLEXIBLE GRAMMAR DESIGN: Unlike PS Grammar, Semantic Grammars do not adhere to any prescriptive framework and can be easily reformed for disambiguation.

Despite their abundance in the corpus, PPs are restricted to locative phrases where the prepositions have constant senses. Moreover, the SL always has null TL. Therefore prepositional ambiguity is negligible. In addition, while pronouns are rare in the corpus, referential ambiguity is also negligible.

(2) PP-Attachment Problems (Structural Ambiguity)

The problem rests on noun PPs and verb PPs, mainly in transitive VPs. e.g.

[1] An active trough of low pressure brought disturbed weather to the coastal areas of Guangdong yesterday.
[2] Afternoon temperature in the region of 30 degrees were recorded over various parts of the territory.

In [1], it is unclear whether the to-phrase modifies the noun "weather" or the verb "brought". In [2], it is unclear whether the of-phrase accompanies the noun "parts" or the verb "recorded". To tackle this problem, three methods have been used:

(a) Subcategorization - "brought" is marked as a ditransitive

verb which must occur in the pattern " brought X to Y".

(b) Modeling common usage - preferences are assigned to the prepositions to mark whether they usually follow noun or verb. Since "of" has the [np] feature, it is believed to follow the noun.

(3) Semantic Grammars: The flexibility of semantic grammar enables it to be easily modified. PPs can be categorized into separate semantic classes or phrases according to their functions so that they will not "clash" together to cause ambiguity. But drawing semantic grammars from complete sentences is complicated, requiring detailed analysis of the sentence structure.

As PP-attachment problem is not serious in the corpus (since most verbs are intransitive and most PPs are sentence PPs), these methods are good enough for disambiguation. But method (b) is ad hoc and insecure. Method (a), though more general, may be risky -- even if a verb has no PP complement, it can have any number of PP adjuncts! So other methods must be used to handle the problem on further extensions.

(3) Verb-Based Transformation

Some of the complete sentences require special transfer formats. So we determine the transfer pattern by the verb type by classifying the verbs according to how they affect structural transfer: TYPE 1 - common verbs, TYPE 2 - passive voice, TYPE 3- "expected" in "is expected to" , TYPE 4 - ditransitive verbs.

(4) Pragmatic considerations

Despite the absence of formal pragmatic processing, there is a mechanism checking the referencing status of the definite determiner "the" for picking correct TLs: nouns with exophoric referents (e.g. the coast, the airport) are classified as "definite nouns". If a noun has a modifying PP and is neither

a proper noun nor a "definite noun", it must have no preceding referent since it has to be specified by the PP. On the contrary, a definite noun, though not mentioned previously and not accompanied by a pp, is assumed to have exophoric referents and is treated as old information.

e.g. 1. Some thundery showers brought nearly 40 millimetres of rainfall to that region.
(old information - has no post-modifying PP
- is not a pn or n_def)

2. A monsoon prevailed over the coastal areas of Guangdong.
(new information - has post-modifying PP for specifying
- is not a pn or n_def

*3. Pressure is low over the Western Pacific to the east of the Philippines.
(old information - it is a n_def, so the following PP is not restrictively specifying it!)

4. Strong gusts were recorded at the airport.
(old information - it is a n_def)

5. Showers continued to affect the territory yesterday.
(old information - it is a proper noun)

(4) Adj-PP attachment problem

ADJs and PPs are both pre-modifiers of NP in the TL. How should they be ordered? Results of corpus analysis reveal that PPs containing a proper noun come before ADJs, or else after it.

(5) Adv-attachment problem

Whether an adverb modifies the whole VP or the verb itself determines the transfer formats. This ambiguity is tackled by isolating the 2 types and checking whether there is a verb-modifying adverb immediately after the main verb.

(6) Adv-grouping problem

In a compound adverb phrase, the SL order of ADVs may not be

the same as the TL one. e.g.

Eng: early yesterday morning
1 2 3

Chi: zuo2tian1 zao3shang chu1qi1
2 3 1

So semantic features are used to mark the ordering preference:

1. [manner] : gradually,generally,much
2. [place] : locally
3. [t1] : yesterday,today
4. [t2] : morning,afternoon
5. [t3] : early

(*NOTE: 1-> 5 decreasing preference)

5.3. Translation Quality

The translation quality is satisfactory except in 2 cases: (1) unusual, especially literary, styles are used, (2) while English is subject-prominent and Chinese is topic-prominent, there is sometimes a strong discrepancy in the topic-comment patterns of the SL and TL.

e.g. Some morning showers brought more than 10 millimetres of rainfall to the territory.

MT: Ji3zhen4 zao3shangde zhou4yu3 wei4 Ben3gang3 dai4lai2le shi2 hao2mi3 de yu3liang4.

Human: Zao3shang you3 ji3zhen4 zhou4yu3, wei4 Ben3gang3 dai4lai2le shi4 hao2mi3 de yu3liang4.

The SL and TL topics are "showers" (zhou4yu3) and "morning" (zao3shang) respectively. With the absence of topic analysis in the system, the difference cannot be identified so that "showers" remains as the topic and "morning" remains as the adjective in the MT version -- resulting in Europeanization. On further extension, a topic analysis component can be added to tackle the

problem, which will require a knowledge base to incorporate world knowledge and Chinese linguistic knowledge to detect topic conversion between English and Chinese.

5.4. FUTURE DEVELOPMENT

There can be further utilizations of the semantic patterns by going beyond the sentence level to discourse -- with a Text Grammar (Grishman, Kittredge 138). The idea is to represent discourse patterns in the grammar and parse the passage as a single unit, which enables more systematic analysis of the text. This is especially useful since we depend on the textual organization of sections in triggering grammar paths.

Grammar rules are now implemented as Prolog clauses and fired directly. If the system is to be enlarged, we had better separate them from the processing algorithms as data, so that new rules can be integrated independently (Picken 85).

Assistant facilities such as a lexicon updating component can be integrated. There can also be a dialog manager which asks the user to update unknown words interactively, which promotes better machine-human cooperation in producing better results.

6. CONCLUSION

The project marks another attempt of fully automatic translating in a restricted domain. The system produced is experiental but operational. New grammars, partitioned parsing, and various methods were put forward as an attempt to improve processing efficiency. Moreover, the sublanguage model constructed leaves a hint to future research in the same domain.

REFERENCES

Allen, James. Natural Language Understanding. London: Benjamin Cummins Publishing Company, 1987.

Bourbeau, Laurent and John Legrberger. Machine Translation. Philadelphia: John Benjamins, 1988.

Chen, Hsin-Hsi and Chen Kuang-Hua. "Attachment and Transfer of Prepostional Phrases with Constraint Propagation." Computer Processing of Chinese and Oriental Lanugages. 6.2 (1992): 123-142.

Grishman, Ralph and Richard Kittredge. Analyzing Language in Restricted Domains: Sublanguage Description and Processing. London: Lawrence Erlbaum Associateds, 1986.

Hutchins, W. J. Machine Translation: Past, Present, Future. Chichester: Ellis Horwood, 1986.

Kay, Martin. "Machine Translation." American Journal of Computational Linguistics. 8.2 (1982): 74-78.

King, Margaret. Machine Translation Today. Edinburgh: Edinburgh UP, 1987.

Kittredge, Richard and John Lehrberger. Sublanguage: Study of Languague in Restricted Semantic Domains. New York: Walter de Gruyter, 1982.

Krulee, Gelbert.K. Computer Processing of Natural Language. London: Prentice Hall, 1991.

Lauren, Christer and Marianne Nordman. Special Language: From Human Thinking to Thinking Machines. Britain: Multilingual Matters, 1989.

Nirenburg, Sergei. Machine Translation: Theoretical and Methodological Issues. Cambridge: CUP, 1987.

Pereira and Shieber. Prolog and Natural Language Analysis. Menlo Park: CSLI, 1987.

Picken, Catriona. Translating and The Computer 8: A Profession on the Move. London: Aslib, 1987.

Smith, George W. Computer and Human Language. New York: OUP,1991.

Steel, Brian D. LPA Prolog Professional Compiler V2.5. London: Logic Programming Associates, 1988.

Williams, Stephanie. Humans and Machine. Norwood: Ablex Publishing Corporation, 1985.

APPENDIX 1

天氣概況：

尋日朝頭早幾陣局部地區性雷雨影響本港，但係雲層逐漸變得輕薄，下晝帶嚟晴朗嘅天氣。喺西太平洋，尋日有一股熱帶低氣壓增強咗成為一股強烈嘅熱帶風暴，個名叫做艾里。 尋晚11點，艾里喺馬尼拉以東大約560公里結集，估計會增強，以每小時大約30公里嘅速度向西北偏西移動。估計艾里會迅速橫過菲律賓，禮拜日進入南中國海。

本地天氣預測：（11/7/1992 禮拜六）

天晴。
氣溫喺26同31度之間。
吹和緩嘅東南風。

禮拜日天氣展望：

初時天晴，晏啲漸漸會好大風，有幾陣雨。

澳門地區今日天氣預測：

多雲，有局部地區性嘅驟雨，日間間中會有陽光。
氣溫喺25同31度之間。
吹輕微至和緩嘅東至東南風。

C:\>_

天氣概況：

尋日下晝初時幾陣局部地區性同埋強烈嘅雷雨影響西貢，為嗰個地區帶嚟咗大約40毫米嘅雨量。而家，有一道高壓脊由太平洋向西面移動，估計禮拜六會影響廣東沿岸地區。尋晚有一個西太平洋低壓區增強咗成為一股熱帶低氣壓。尋晚11點，喺馬尼拉以東大約1240公里結集，估計會以每小時大約22公里嘅速度向西面移動。

本地天氣預測：（10/7/1992 禮拜五）

除咗有幾陣驟雨，大致上都係天晴。
氣溫喺26同31度之間。
吹和緩嘅南至東南風。

禮拜日天氣展望：

禮拜六天晴同埋炎熱。

澳門地區今日天氣預測：

初時大致上都係多雲，有局部地區性嘅驟雨，晏啲漸漸會天晴。
氣溫喺27同32度之間。
吹輕微至和緩嘅南至西南風。

C:\>_

Toward
Discourse-guided Theta-grid Chart Parsing for Madarin Chinese
-- A Preliminary Report

Koong H. C. Lin and Von-Wun Soo
Department of Computer Science, National Tsing-Hua University HsinChu,
Hsinchu, 30043, Taiwan, R.O.C.
E-Mail:soo@cs.nthu.edu.tw

Abstract

An attempt for this work is to show a way of combining word identification, syntactic processing, semantic processing, and discourse processing into a cohesive framework. We utilize thematic information as a media to integrate these processing modules. In this work, the thematic information is assumed to reside in lexicons based on the theta grid theory. For determining the main verb(s) of a sentence with Serial Verb Constructions (SVCs), we propose an algorithm which evaluates a scoring function; examples showing how Chinese texts with SVCs in the legal domain can be parsed by our parser are presented. We also show how the previously acquired anaphoric knowledge can be used to guide the chart parser.

1 Introduction

Traditional natural language processing (NLP) systems are normally composed of many standing alone modules to perform individually and sequentially the word identification, the syntactic processing, the semantic processing, and the discourse processing. However, problems such as PP-atachments, anaphora, and structural ambiguities cannot be easily resolved if these modules are not cohesively interacted with each other. Thus, some attempts were made to integrate these modules by enhancing interactions between modules, or making the boundaries between modules vague. *Preference semantics* [Wilks75] [Fass83], *case-based parser* [Martin89], *expectation-driven partial parsing* [Rau87], and *conceptual parsing* [Shank73], were paradigms of such attempts. In this work, we also aim at integrating these modules into a cohesive framework, in which thematic information plays a significant role. We also make use of the anaphoric knowledge acquired by our previous work, *G-UNIMEM* [Chen92], to "*predict anaphora*" during parsing.

A sketch of our work is illustrated in [figure 1]. The functions of each module are briefly described as follows: the *TG-Chart parser*, which accepts the input from the word identifier, and interacts with the Discourse Daemon, is the core of our work. Syntactic knowledge in grammar rules, and thematic information in lexicons, are the two major knowledge used by TG-Chart parser. The *partial word identifier* partially identifies the input sentence, and constructs a word lattice, which serves an input for the parser. The most interesting module, the *Discourse Daemon*, utilizes a set of G-Rules to make prediction for occurrences of anaphora, where G-Rules are acquired by G-UNIMEM. More precise descriptions of each module will be given in the following sections.

This work serves as a natural language front-end for our long-term research of a verdict understanding system. Thus, the corpora we use are some judicial verdict documents from the Kaohsiung district court [臺90a] [臺90b], which were written in a special official-document style. Thus, our analysis is based on such a kind of sub-language.

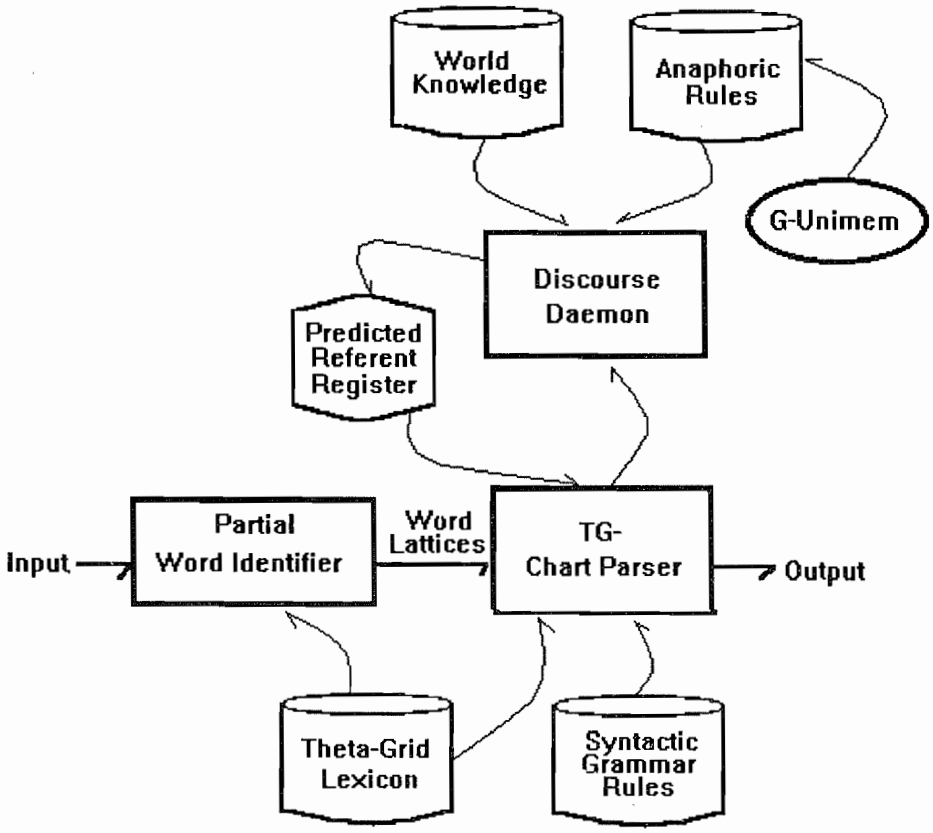


Figure 1. A schematic framework of Discourse-guided Theta-grid chart parser

2 The TG-Chart Parser

2.1 The Theta-Grid Theory and the Chart Parser

Thematic information is one of the information sources that can bridge the gap between syntactic and semantic processing phases. Tang proposed a *theta-grid theory* [湯 92] in which rich thematic information is incorporated for the analysis of human languages. The idea of theta-grid theory is as follows: we use a predicate, say, a verb, as the *center* of a "grid" and, by finding the theta-roles registered in the lexical entries of this predicate, we can construct a grid formed by this predicate and then construe the sentence (or clause) spanned by this predicate. As we know, Chinese is not sensitive to syntax; therefore, the theta-grid seems to be suitable for processing Chinese. However, to computationalize theta-grid theory, some control strategies for parsing must be included.

The well-known *chart parser* [Kay80] [Allen87], which utilizes a data structure called "*chart*" to record the partial parsing results, is suitable for our work. Since it considers all possible combinations of constituents, it is more flexible and can accept sentences with missing theta roles. Thus, we design a modified chart parser called TG-Chart parser by combining the theta-grid theory and the chart parser. Note that currently in our work, only the theta grids for *verbs* are considered. For each verb, there are two kinds of theta roles registered: the *obligatory roles*, which *must* be found for this verb to construct a legal "grid"; the *optional roles*, with their appearance being optional. Take "告訴" for example, its theta roles are registered as: *+[Th (Pd) Ag]*; thus, two *NPs* must be found in the chart for the construction of a legal grid (From *syntactic clues*, both *"Ag"* and *"Th"* are always played by *NPs*

according to Liu. [Liu93].), while the appearance of a clause to serve as a "Pd" role is optional. Besides, some theta roles, like Qd, Ma, Ti,..,etc., are not registered in the lexical entries of individual verbs, since they occur commonly in grids formed by every kinds of verbs.

A brief description of our parsing algorithm is as follows:

[Step 1] Search the sentence for positions of all "possible verbs". (what we call *possible verbs* are those words with *verb*-category as one of its syntactic categories)

[Step 2] By considering all possible combinations, the chart parser groups the words into *syntactic constituents*. Syntactic knowledge is used in this step.

[Step 3] If only one verb is found in [Step 1], search the chart for constituents which can play the theta roles of this verb.

[Step 4] If more than one verb are found, more complex considerations are needed. We will discuss such a situation more detailedly in the next section.

Note that currently in our work, only simple information is encoded in the lexicon. Thus, we need a small set of syntactic grammar rules, and a syntactic chart parser to group the phrases. While the lexicon is enlarged and enriched, it seems better to drop the syntactic grammar rules, and drive our parser toward *information-based* and *unification-based*. The successful ICG parser [Chen 89] [Chen 90] is a paradigm for our further development.

2.2 Dealing with Serial Verb Constructions

Serial verb construction (SVC) is a unique construct of the Chinese language, which refers to a sentence containing two or more VPs juxtaposed without any marker indicating what the relationships are between verbs [Li81]. Many works are reported on processing different sorts of SVCs. Some of them are rule-based [Chang91], some are lexicon-driven based on Case Grammar [Yeh92] [Pun91] [Fillmore68]. Linguists classified SVCs into five types: *two and more separate events*, *pivotal construction*, *descriptive clauses*, *sentential subjects*, and *sentential objects*. This classification is the basis for their analysis. In our legal domain corpora, there are also many occurrences of SVCs. Since our parser is based on the theta grids, in case of SVCs, different verbs will *compete* in finding their own theta roles. Thus, some mechanism for arbitrating among verbs for the ownerships of each constituent in the chart must be designed. According to Yorick Wilks, *language does not always allow the formation of "100%-correct" theories* [Hirst81]; therefore, we attempt to find a more flexible method for recognizing SVCs. We propose a *scoring function* to select a "preferable" construction for the sentence with SVCs. The scoring function is defined as follows, where RWR is the abbreviation of "Ratio of Words included in some phrase with Roles assigned", OBR, "OBligatory Role", and OPR, "OPtional Role" (Note that OBR and OPR indicate those roles *registered in theta grids*.):

$$Score = \frac{\sum_{\text{every verb}} Score - Per - Verb}{\text{number of verbs}} \quad (3.3.1)$$

$$Score - Per - Verb = \frac{[(\text{number of OBR found}) * 2 + (\text{number of OPR found})]}{Base} * RWR \quad (3.3.2)$$

$$RWR = \frac{\text{number of words included in some phrase with roles assigned}}{\text{number of words in the clause}} \quad (3.3.3)$$

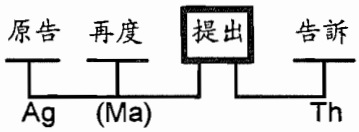
$$Base = 2 * (\text{number of OBR}) + (\text{number of OPR}) \quad (3.3.4)$$

The score is calculated as the average value of scores obtained by each verb in the sentence (as in equation 3.3.1). For each verb, the score is estimated by two factors: *first*, the ratio of theta roles found, and, *second*, the ratio of words with roles assigned , i.e., RWR. For precise calculation, see equation (3.3.2). We heuristically weigh the relative significance between obligatory roles and optional roles by *2:1*, as in (3.3.2) and (3.3.4). In some cases, the verb finds many theta roles in the clause it constructs, but the words in this clause are not all assigned roles. We consider such assignment doesn't construe the real construction of the sentence. Thus, to reflect such cases, we calculate RWR by dividing the number of words which are included in some phrase with a role assigned by the total number of words in the clause (see equation 3.3.3). Now, let's illustrate the calculation of this scoring function by the following examples:

【Example 1】"原告 再度 提出 告訴"

In [Step 1], "提出" and "告訴" are both found as "possible verbs". Here "告訴" has two syntactic categories registered in its lexical entry: verb and noun, while "提出" has only one categorty, the verb. The theta grid for "提出" is +[Th Ag], "告訴", +[Th (Pd) Ag]. So, to decide whether "告訴" is treated to a verb or a noun, there are four cases to be considered:

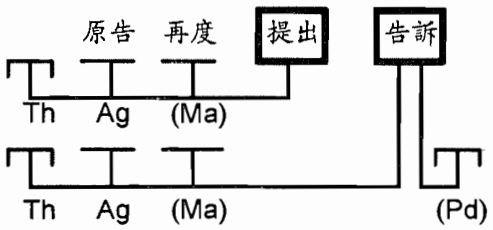
(1) "提出" is treated as a verb, while "告訴" a noun.



In the above, "提出" enveloped by a *box* means it plays a verb. When it searches for theta roles, "原告" and "告訴" are respectively found as its Ag and Th, the two obligatory theta roles registered in its lexical entry. In addition, "再度" is found as an optional role, Ma. However, Ma is not registered in the lexical entry of "提出", it contributes no credit for "提出". Now, the score is calculated as follows:

For "提出", there are two obligatory roles, so Base = 2*2 = 4. Moreover, in this sentence, "原告","再度","提出",and "告訴" are all assigned some roles; thus, RWR = 4/4 = 1. And then, Score-Per-Verb = {[(number of OBR found)*2 + (number of OPR found)]/Base} * RWR = {[2*2+0]/4}*1=1. Finally, Score = 1/1 = 1.00.

(2) "提出" and "告訴" both are treated as verbs.

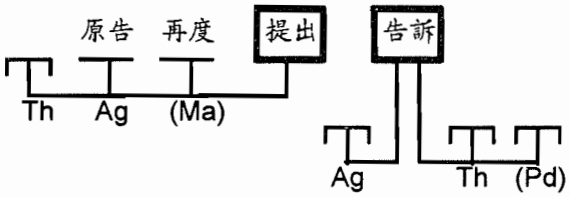


In the above figure, "提出" cannot find its Th, "告訴" cannot find its Th and Pd. Such "*cannot find*" situations are represented by the symbol "┌┐".

For "提出", Base=4. Note that for the portion of sentence centered by "提出", "原告 再度 提出", every word is assigned a role; thus, RWR = 3/3 = 1. Score-Per-Verb = {[1*2]+0]/4}*1=0.5.

For "告訴", Base=2*2+1=5. RWR = 3/3 = 1. Score-Per-Verb = {[1*2+0]/5}*1=0.4. So, for this case, Score = (0.5+0.4)/2 = 0.45.

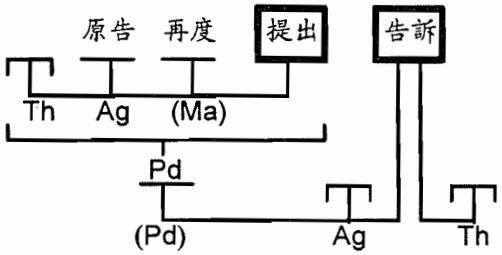
(3) "提出" and "告訴" both are treated as verbs, while "告訴" is subordinated to "提出".



For "提出", Base=4. Since "告訴" is subordinated to "提出", but the clause it forms cannot play any role for "提出", the RWR for "提出" is 3/4 = 0.75. Score-Per-Verb = {[1*2+0]/4}*0.75 =0.375.

For "告訴", it is clear that Score-Per-Verb is 0, because it cannot find any role. So, Score = (0.375+0)/2 = 0.1775

(4) "提出" and "告訴" both are treated as verbs, while "提出" is subordinated to "告訴".



For "提出", Base=4. RWR=3/3=1. Score-Per-Verb = {[1*2+0]/4}*1 = 0.5. The clause constructed by "提出" supports a Pd role for "告訴". Thus, for "告訴", RWR=4/4=1; Score-Per-Verb = {[0*2+1]/5}*1 = 0.2.

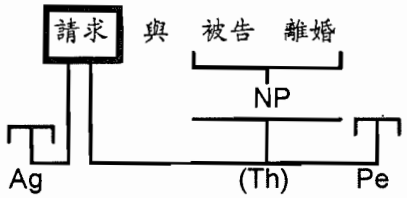
Score = (0.5+0.2)/2 = 0.35.

From the above discussions, **case(1)** apparently gets the highest score (1.00). So, the parsed structure in case(1) is preferable to those in the other cases. That is, in this sentence, "提出" plays as the only verb, while "告訴" plays a noun. Therefore, the right syntactic category for "告訴" in this sentence is determined.

【Example 2】"請求 與 被告 離婚"

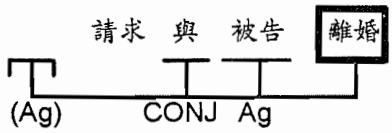
In [Step 1], "請求" and "離婚" are both found as "possible verbs". Here "請求" and "離婚" both have two syntactic categories registered in its lexical entry: the verb and the noun. The theta grid for "請求" is +[(Th) Pe Ag], "離婚" +[Ag (Ag)]. So, there are five cases to be considered:

(1) "請求" plays as the only verb, while "離婚" plays as a noun.



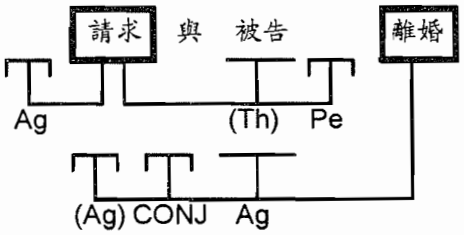
For "請求", Base=5. Since "與" can't play any role in this sentence, RWR = 3/4 = 0.75. Score-Per-Verb = {[0*2+1]/5}*0.75=0.15. So, Score = 0.15/1 = 0.15.

(2) "離婚" is treated as a verb, while "請求" a noun.



For "離婚", Base=3. Note that although "請求" is an NP, it cannot play as Ag for "離婚". It is because it doesn't satisfy the constraint for playing as Ag: an Ag must has a feature "+animate", according to Gruber's theory that an agent must be *an entity with intentionality* [Gruber76]. So, RWR = 3/4 = 0.75, and Score-Per-Verb = {[1*2+0]/3}*0.75 = 0.5. Score = 0.5/1 = 0.5.

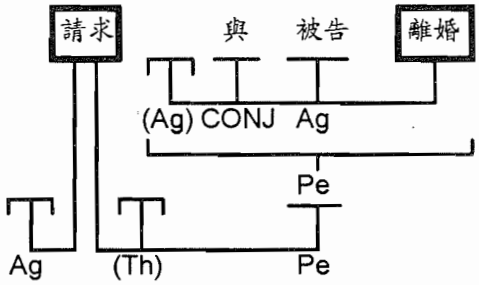
(3) "請求" and "離婚" both are treated as verbs.



For "請求", Base=5, RWR=2/3=0.67, since "與" doesn't play any role. Score-Per-Verb = {[2*0+1]/5}*0.67=0.134.

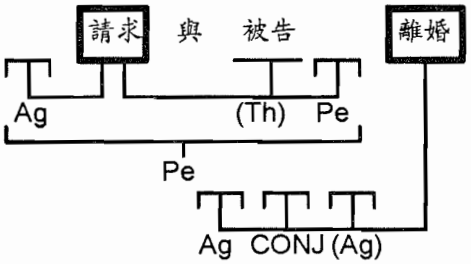
For "離婚", Base=3. RWR=3/3=1. Score-Per-Verb = {[1*2+0]/3}*1 = 0.67. Score = (0.134+0.67)/2 = 0.402.

(4) "請求" and "離婚" both are treated as verbs,with "離婚" being subordinated to "請求"



For "請求", Base=5. RWR=4/4=1. Score-Per-Verb = {[1*2+0]/5} = 0.4.
For "離婚", Base=3. RWR=3/3=1. Score-Per-Verb = {[1*2+0]/3} = 0.67.
Score = (0.4+0.67)/2 = 0.535.

(5) "請求" and "離婚" both are treated as verbs,with "請求" being subordinated to "離婚".



For "請求", RWR=2/3=0.67. Score-Per-Verb = {[2*0+1]/5}*.0.67 = 0.134.
For "離婚", it's clear that Score-Per-Verb = 0.
Score = (0.134+0)/2 = 0.067.

From the above discussions, **case(4)** apparently gets the highest score (0.535). So, the parsed structure in case(4) is preferable to those in the other cases. That is, in this sentence, "請求" and "離婚" both are treated as verbs, while "離婚" is subordinated to "請求". The clause constructed by "離婚" is assigned the Pe role for "請求". It is one kind of Serial Verb Construction. (This kind of SVC is commonly called "*sentential objects*".)

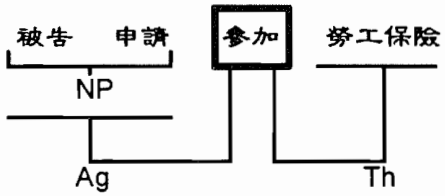
In table 1, we show the results of more sentences with SVC in the legal documents which are parsed by this scheme in our TG-Chart parser. The sample sentences are as follows:

S1: 原告 訴請 被告 給予 三十萬元
S2: 原告 請求 被告 清償 債務
S3: 被告 未 到 場 爭執
S4: 被告 於 民國七十八年 十一月二十日 突 無故 離家 出走
S5: 被告 未 返 家 與 原告 同居
S6: 原告 聲請 訊問 證人
S7: 被告 希望 原告 能 諒解
S8: 被告 申請 參加 勞工保險
S9: 原告 通知 被告 改善
S10: 原告 本人 到 場 對質
S11: 被告 否認 有 過失
S12: 原告 訴請 被告 負擔 訴訟費

Sen. No.	Verb Candidates	Verb Players	Relatioships	Highest Score	Correctness
S1	v1: 訴請 v2: 給予	v1,v2	v1>v2	1.00	Y
S2	v1: 請求 v2: 清償	v1,v2	v1>v2	1.00	Y
S3	v1: 到 v2: 爭執	v1,v2	v1=v2	1.00	Y
S4	v1: 離家 v2: 出走	v1,v2	v1=v2	1.00	Y
S5	v1: 返 v2: 同居	v1,v2	v1=v2	0.83	Y
S6	v1: 聲請 v2: 訊問	v1,v2	v1>v2	0.70	Y
S7	v1: 希望 v2: 諒解	v1,v2	v1>v2	0.84	Y
S8	v1: 申請 v2: 參加	v2		1.00	N
S9	v1: 通知 v2: 改善	v1,v2	v1>v2	0.83	Y
S10	v1: 到 v2: 對質	v1,v2	v1=v2	0.88	Y
S11	v1: 否認 v2: 有	v1,v2	v1>v2	0.75	Y
S12	v1: 訴請 v2: 負擔	v1,v2	v1>v2	1.00	Y

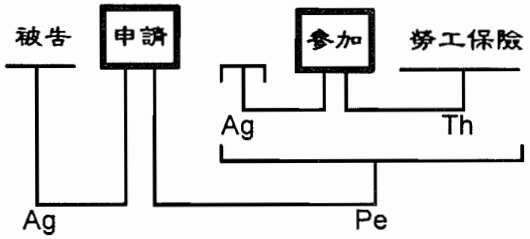
Table 1. More sample sentences with SVCs

The notation "v1>v2" means v2 is subordinated to v1, "v1=v2", no subordination relations exist between verbs.. In the above table, the result for S8 is incorrect. Analyzing this sentences, we find that it is caused by the *incorrect formation of compound nouns*. In S8, both "申請" and "參加" are the *possible verbs*, while "申請" has two syntactic categories: the verb and the noun. When "參加" is treated as the only verb, while "申請" a noun, the combination is as follows:



"被告" and "申請" incorrectly combine together and form the incorrect compound noun: "被告申請". Thus, "參加" finds "被告 申請" and "勞工保險" as its Ag and Th, respectively; and, moreover, obtains a high score: 1.00.

However, we know that the correct combination should be as follows, where "申請" and "參加" are both treated as verbs:



It obtains the *secondly* high score: 0.75, although it is the *correct* construction. Thus, a study for the *Serial Noun Constructions*, such as the work reported by Yeh et al. [Yeh91], is one of our future works.

3 The Discourse Daemon

3.1 G-Rules

The *anaphora* problem plays a significant role in natural language processing systems. The *raise-bind mechanism* [Lin86], which was based upon the *empty categories*, supports a mechanism for resolving *intra-sentential* anaphora. Many problems arise during discourse processing. However, currently in our work, we forcus on the resolution of *inter-sentential* anaphora problems only. In discourse, there may be anaphora in two consecutive sentences [Li86]. Examples of anaphoric ambiguities will be shown later. When anaphora appear in a pair of consecutive sentences, the two consecutive sentences are called *conjoined sentences*. In our work, the Discourse Daemon is just the module which resolves the anaphora problems in conjoined sentences. In Anaphora Daemon, two kinds of knowledge may be used: (1) *Anaphoric rules* generated by G-UNIMEN. (2) *Domain knowledge* However, in this paper, only anaphoric rules are reported. In the following, we concentrate on the usage of anaphoric rules.

The anaphoric rules used by Discourse Daemon are called *G-Rules*. We use G-Rules to predict and resolve the anaphora occurring in conjoined Chinese sentences. Here we show some sample G-Rules following:

1. [ante(agent), type(nil)]
   :- [f1(agent), anaphor(agent)]
2. [ante(theme), type(nil)]
   :- [f1(theme), anaphor(theme)]
3. [ante(agent), type(nil)]
   :- [f1(agent), f2(theme), anaphor(agent)]
4. [ante(agent), type(pronoun)]
   :- [f1(agent)]
5. [ante(theme), type(pronoun)]
   :- [f2(theme)]
6. [ante(theme), type(pronoun)]
   :- [anaphor(theme), f1(agent), f2(theme)]

G-Rules are written in a Prolog-like style. For rule 1, it means that if in the first sentence, one word plays the *agent* role (represented by "*f1(agent)*"), and in the second sentence, an *anaphor* occurs in the *agent position*, this anaphor will refer to the word which is in the *agent position* of the first sentence. (i.e., *the antecedent of this anaphor is the agent in the first sentence*.) In addition, the surface representation of this anaphor is *zero-pronoun*. (This is just what the notation "*type(nil)*" means.) We can use the following sentence to illustrate such rules:

[他]i 跌倒了, []i 很難過.
agent agent

In the first sentence, "他" can play the only role in this sentence as an agent; and, in the second sentence, an anaphor occurs at the agent position, represented as zero-pronoun. Thus, by rule 1, we know that this anaphor refers to "他" in the first sentence.

Similarly, we can use G-Rules to analyze another example:

[老張]i 娶了一個 [美嬌娘]j, []i 快樂極了.
agent theme agent

In the first sentence, "老張" and "美嬌娘" are treated as agent and theme respectively; and, in the second sentence, an anaphor occurs at the agent position, also represented as zero-pronoun. According to G-Rule 3, the antecedent of this anaphor is likely to be "老張", the agent of the first sentence. It is obviously a correct choice. From the above two examples, we find that G-Rules can be used to *resolve* and, besides, to *predict* the antecedents of anaphora occurring in conjoined Chinese sentences. In the next section, we will see how to utilize G-Rules in anaphora prediction.

3.2 Use G-Rules to Predict Anaphora

Let's observe an example directly, here a pair of sentences are conjoined:

原告再度提出告訴，請求與被告離婚

Due to the analyzed result in section 3.3, we know that in the first sentence, "提出" is treated as the only verb, while "原告" as its agent, and "告訴" as its theme:

[原告]i 再度提出 [告訴]j
Agent Theme

Search G-Rules for matched rules, we find that such situation in this sentence satisfies the first two conditions for rule 3:

```
[type(nil), ante(agent)]
:- [f1(agent), f2(theme), anaphor(agent)]
```

That is, there are both roles of agent and theme in the first sentence, which satisfies the f1 and f2 conditions in rule 3 respectively. Thus, if there is an anaphor which appears as zero-pronoun in the second sentence to be parsed later, the antecedent should be the agent in the first sentence, i.e., "原告". So, "原告" *is kept in a temporary register, Pred-Ref*, since it is likely to be the referent of the next sentence.

And then, when the second sentence is parsed (See the analysis in section 3.3, case(4) of example 2.), exactly as we expected, there is an anaphor at the agent position, with this agent ommitted:

[]i 請求 [Pe []i 與 [被告]j 離婚]
agent agent agent

So, "原告" kept in Pred-Ref is extracted to fulfill the agent position. And, moreover, "原告" continues propagating to the embedded clause, i.e., the Pe "與被告離婚", also fulfills the ommitted agent. (The theta role Pe means a proposition without a subject.) Therefore, we get a completely parsed sentence pair as follows:

[原告]i 再度提出 [告訴]j, [原告]i 請求 [原告]i 與 [被告]k 離婚
agent theme agent agent agent

Thus, the predictive power supported by G-Rules is exhibited.

4 Discussions

We have proposed a framework in which each component for natural language processing can be cohesively correlated. The Discourse Daemon, a module which utilizes a set of G-Rules, makes TG-Chart parser anaphora predictable. In our current status, the parser is implemented in C language. We believe it quite easy to incorporate Discourse Daemon into the parser. G-UNIMEM is well implemented, and gets accuracy rates 95.8% for resolving and 90.8% for choosing anaphora with 120 training instances. The reported accuracy supports the reliability for G-Rules.

There are some concerns in the further development of our system:

(1) The pre-defined feature set used to train G-UNIMEM could be extended, since in some cases these features do not seem to be adequate. Observe the following sentence:

[老張]i 娶了一個 [老婆]j, []j 很會做飯
agent theme agent

An anaphor occurs in the second sentence. It is obvious that the antecedent of this anaphor is "老婆", the theme in the first sentence. However, if we apply our G-Rules, "老張" will be selected due to rule 3, cause an incorrect prediction. We can easily see that such a link between this anaphor and its antecedent is caused by the relation: "老婆" is always the agent for "做飯" when we recall our previously seen cases. So, we think that this problem can be resolved either by adding such semantic features or applying world knowledg. How to integrate world knowledge is our furture work.

(2) There are two problems which might happen during the application of G-Rules: First, the G-Rules might not be 100%-accurate. Second, There might be cases where two possible predictions can be invoked by two G-Rules for the same sentence. Thus, more careful considerations are needed for the design and the application of G-Rules. In addition, how to

extend our work so that it could handle anaphora occurring in sentences that are *not simple conjoined sentences*, is also one of our concerns.

(3) In Madarin, either an NP, a PP, or a S can play a role. However, in our work, the possibility of a PP is not considered yet. It's also our future concern.

(4) Previous works always treate word identifier as a preprocessor of the subsequent parser. However, in the analysis of the corpora of the verdict documents, we found it was so complicated that it was impractical to isolate the word identifier to be a standing alone module. So, we attempt to integrate the word identifier and the parser into a cohesive one. That is, the word identifier finds coarse word boundaries first, and the parser refines the boundaries and produces the final correct word identifications.

Word lattices are often used in speech recognition and word identification [Lee91] [Carter92], since the word boundaries for the input always are not definite. Lee et al. found that the data structure, chart, used by chart parser, is suitable to represent such word lattices, and utilize a chart parser to parse the indefinite input, a word lattice, which are produced during the speech recognition module, to get a correct recognition. We find that the situation is similar to our work. So, we imitate the idea and propose a two-phase algorithm which will be developed in the future. Our algorithm is quite simple: In phase 1, for each character C in the sentence, word identifier searches the lexicon for all words that beginning by C, and keep the matched word to construct a word lattice. In phase 2, TG-Chart parser parses this word lattice and gets a correct identification.

As an example, for following sentence, it is not reasonable for a standing alone word identifier to achieve a 100%-correct identification:

"持其所有平日做皮雕工作使用之槌子一把"

In phase 1, word identifier constructs a partially identified word lattice:

In phase 2, TG-Chat parser is used to parse this lattice. We can correctly select the following identification, since the other possibilities will be filtered out during parsing:

持 其 所 有 平日 做 皮雕 工作 使用 之 槌子 一把

Thus, by combining the word identifier and the parser together, we think a better result will be obtained.

Acknowledge

We would like to thank Benjamin L. Chen, the G-UNIMEM he developed provides us useful knowledge for predicting anaphora.

References

[Allen87] James Allen, Natural Language Understanding, The Benjamin / Cummings Publishing Co. 1987.

[[Carter92] David M. Carter, Lattice-Based Word Identification in CLARE. In Proc. of 30th Annual Meeting of Association for Computational Linguistics (ACL-92).

[Chang91] Chao-Huang Chang and Gilbert K. Krulee, Prediction Ambiguity in Chinese and Its Resolution. Proc. of ICCPCOL 1991, pp.109-114.

[Chen89] 陳克健,黃居仁，訊息爲本的格位語法---一個適用於表達中文的語法模式，ROCLING II, 1989.

[Chen90] Keh-jiann Chen and Chu-Ren Huang, Information-based Case Grammar. In Proc. of COLING-90.

[Chen92] Benjamin L. Chen and Von-Wun Soo, An Acquisition Model for Both Choosing and Resolving Anaphora in Conjoined Madarin Chinese Sentences. In Proc. of COLING-92, pp. 274-279.

[Fass83] Dan Fass and Yorick Wilks, Preference Semantics, Ill-Formedness, and Metaphor. American Journal of Computational Linguistics, Vol.9, No.3-4, pp.178-187. 1983.

[Fillmore68] Fillmore, C., The Case for Case. In Universals in Linguistics Theory, ed. E. Bach and R.T. Harms. New York: Holt. (1968).

[Gruber76] Gruber J. S., Lexical Structures in Syntax and Semantics, North-Holland Publishing Company. 1976.

[Hirst81] In Graeme Hirst, Lecture Notes in Computer Science, Anaphora in Natural Language Understanding, A Survey. Springer-Verlag Berlin Heidelberg. 1981.

[Kay80] Martin Kay. Algorithm Schemata and Data Structures in Syntactic Processing. In Proc. of the Nobel Symposium on Text Processing, Gothenburg, 1980.

[Lee91] Lin-Shan Lee, Lee-Feng Chien, L.J. Lin, J. Huang, K.-J. Chen. An Efficient Natural Language Processing System Specially Designed for the Chinese Language. Computational Linguistics, 17(4), pp.347-378. 1991.

[Li86] Mei Du Li, Anaphoric Structure of Chinese, Student Book CO., Taipei, Taiwan. 1986.

[Li81] C. N. Li and S. Thompson, Mandarin Chinese: a Functional Reference Grammar, University of California Press, Berkeley. 1981.

[Lin86] Long-Ji Lin, James Huang, K.J. Chen, and Lin-Shan Lee, A Chinese Natural Language Processing System Based upon the Theory of Empty Categories. In Proc. of AAAI 1986.

[Liu93] Rey-Long Liu and Von-Wun Soo, An Empirical Study of T-hematic Knowledge Acquisition Based on Stntactic Clues and Heuristics. In Proceedings of ACL 1993.

[Martin89] Charles E. Martin, Case-Based Parsing. In C. K. Riesbeck and R. C. Schank "Inside Case-based Resoning", Lawrence Erlbaum Associates, Inc. 1989.

[Pun91] K. H. Pun, Analysis of Serial Verb Constructions in Chinese. ICCPCOL 1991, pp.170-175. 1991.

[Rau87] Lisa F. Rau, Knowledge Organization and Acess in a Conceptual Information System. Information Processing and Management. Vol.23, No.4, pp.269-283. Special Issue on Artificial Intelligence for Information Retrieval. 1987.

[Shank73] Shank, R. C., Identification of Conceptualizations Underlying Natural Language. In Computer Models of Thought and Language, ed. R. C. Shank and K. M. Colby. San Francisco: Freeman. 1973.

[Wilks75] Yorick Wilks, An Intelligent Analyzer and Understander of English. In B. J. Grosz, K. S. Jones, and B. L. Webber "Reading in Natural Language Processing". 1975.

[Yeh91] Ching-Long Yeh and Hsi-Jian Lee, Resolution of Serial Noun Constructions in Chinese. In Proc. of ROCLING IV, pp.97-110. 1991.

[Yeh92] Ching-Long Yeh and Hsi-Jian Lee, A Lexicon-Driven Analysis of Chinese Serial Verb Constructions. In Proc. of ROCLING V, pp.195-214. 1992.

[湯92] 湯廷池 (1992), 語法理論與機器翻譯: 原則參數語法, ROCLING V, pp.53-83. 1992.

[臺90a] 臺灣高雄地方法院, 臺灣高雄地方法院刑事裁判書彙編第一冊. 1990.

[臺90b] 臺灣高雄地方法院, 臺灣高雄地方法院民事裁判書彙編第一冊. 1990.

Developing a Chinese Module in UNITRAN *

Zhibiao Wu, Loke Soo Hsu, Martha Palmer, Chew Lim Tan
Department of Information System & Computer Science
National University of Singapore
Republic of Singapore, 0511
E-mail: wuzhibia,hsuls,mpalmer,tancl@iscs.nus.sg

Abstract

This paper will share with the readers our experiences gained in a project of translating Chinese to other languages with Principle Based Machine Translation (PBMT). UNITRAN is a prototype system developed in MIT which translates simple sentences among English, Spanish and German. Based on Government Binding (GB) theory and Lexical Conceptual Structure (LCS) theory, UNITRAN serves a good model for applying GB and LCS to achieve the principle based machine translation. We have tried to put Chinese into the system. Now the system can translate among the four languages properly. In the following sections, we will first introduce the basic idea of PBMT. Then we briefly explain how UNITRAN translates Chinese to English. Our major focus will be the Chinese language parameter setting. Some GB parameters will be discussed in certain detail. And finally, the last section will discuss the merit of the PBMT and problems arise from this approach.

1. Introduction

With the development of Government Binding (GB) theory (Sells, 1985) and Lexical Conceptual Structure (LCS) theory (Jackendoff, 1991), the principle based machine translation (PBMT) has drawn some attention. The basic idea of PBMT is that by highly abstracting the regularities existing in human languages, a large part of the language grammar and lexical semantics can be covered by a small number of principles. Different languages get their particular expressions by setting parameters for these principles. The idea is based on the assumption that human has an innate Universal Grammar which enables one to compose new lexical concepts based on a set of semantic primitives.

UNITRAN is a PBMT machine translation prototype developed in MIT by Bonnie Dorr (Dorr, 1990). The system can freely translate single sentences among English, Spanish and German. UNITRAN elaborates the idea of PBMT to its full strength. At each level of morphological, syntactic and semantic processing, the system is designed based on a small

*Special thanks to Bonnie Dorr for her kind permission to use some of the materials in her PhD thesis.

set of principles. Particular languages realized themselves in the system by a set of parameter setting files in the system. This approach brings a lot of merits. First, it is easy to extend the system's ability to handle a new language. By specifying the parameter files, the major parts of the system remind unchanged. Secondly, Different languages have divergences in syntax, semantics and pragmatics level. Since the divergences of the languages can be represented by different parameters for the same principle, language divergences can be easily resolved.

In order to investigate the strength of UNITRAN and the idea of PBMT, and to see whether Chinese is suitable for a PBMT treatment, we have tried to put Chinese in UNITRAN. In this paper, we will focus on the Chinese GB setting. In Section 2, we will briefly introduce the GB and LCS theory. In Section 3, we will present examples to show how UNITRAN handle the Chinese sentence. The Chinese GB parameter setting will be discussed in Section 4. And finally, the last section will discuss the merit of the PBMT and problems may arise from this approach.

2. GB and LCS

This section will briefly introduce the GB and LCS theory and how they are realized in UNITRAN system.

2.1. Government Binding theory

A detail introduction of GB theory can be found in (Sells, 1985). The Chinese GB theory has been developed by (Huang, 1982), (Li, 1990), and recently, Professor Tang T. C. has done a lot of work on this subject (Tang, 1989; Tang, 1992). Followingly, I briefly summarize the theory.

The basic idea of the theory can be shown in the following figure.

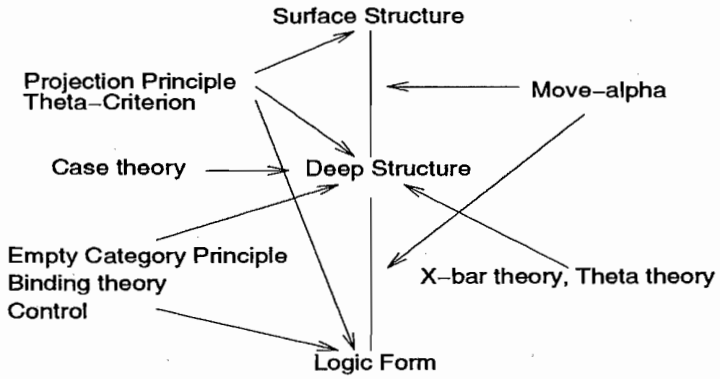


Figure 1: Government-Binding Theory

The language expression is a projection from the lexical semantics. By θ criterion and projection principle, the arguments of a LCS expression can be projected to deep structure to

form word phrase. By 'Moving anything anywhere' (Move-α) with some constraint principles, the surface structure can be derived. Some of the important concepts is explained below:

X-bar theory says that language expressions can be classified into several phrase categories. The syntactic behaviors of each category are similar. For example, in UNITRAN, English is been classified into six categories V, N, A, P, C and I standing for Verb, Noun, Adjective, Preposition, Complementizer and Inflection. Each category follows the rule schemata: For word belongs to a X category, it can form the intermediate phrase X′ with the complement of the X category. The X′ can recursively form a new X′ with the adjunct of the X category. And the X″ i.e. the maximum projection of X is finally formed with specifier and X′. The word phrase is then projected to the surface by Move-α with some constraints like Empty Category Principle, Binding Theory, Control and Case Theory.

θ theory is for the linking from the deep structure to the logic form. θ role is the thematic roles of predicate's arguments. θ criterion says that each argument bears one and only one θ role, and each role is assigned to one and only one argument. This derived the linking rules in UNITRAN.

Projection principle says that representations at each level are projected from the lexicon in that they observe the subcategorization properties of lexical items.

Move-α is an operation from deep structure to surface structure. It means 'Move anything anywhere'. But the moving must obey some constraints. Following are the example of NP-movement and Wh-Movement.

```
1) NP-movement:
   d:    [NP ]  INFL kiss-en Bill
   s:    Bill_i INFL kiss-en e_i

2) Wh-movement:
   d:  [COMP      ] Bill INFL see who
   s:  [COMP who_i] Bill INFL see e_i
```

Bounding says that any application of Move-α may not cross more than one bounding node.

Abstract case: is a notion of NP to show its relation to verbs. NP have cases, verb assigns Accusative Case to its object. Four cases are used in UNITRAN. They are Nominative, Accusative, Dative, and Genitive.

Trace is concerned with the empty position left behind when a constituent has moved by Move-α.

Binding is concerned with the coreference relations among noun phrases. Following sentence show that illegal bindings with the mark $*$.

```
  John_i sees himself_i
* John_i sees himself_j
* John_i sees him_i
  John_i sees him_j
```

2.2. Lexical Conceptual Structure

Lexical conceptual structure is a compositive representation method for lexical semantics. The building blocks can be classified into several types as: EVENT, STATE, POSITION, PATH, THING, PROPERTY, LOCATION, TIME and MANNER. For each type, there is a set of semantic primitives. For example, we have HERE, THERE, LEFT, RIGHT, UP, DOWN ... in the type of LOCATION. Some of the primitives are predicates which takes arguments. According to Jackendoff's theory (Jackendoff, 1991), every sentence meaning or word meaning can be represented by primitives or the composition of primitives. The composition is done by observing the θ theory. Following is an example of Chinese "划伤 (HuaShang)" event.

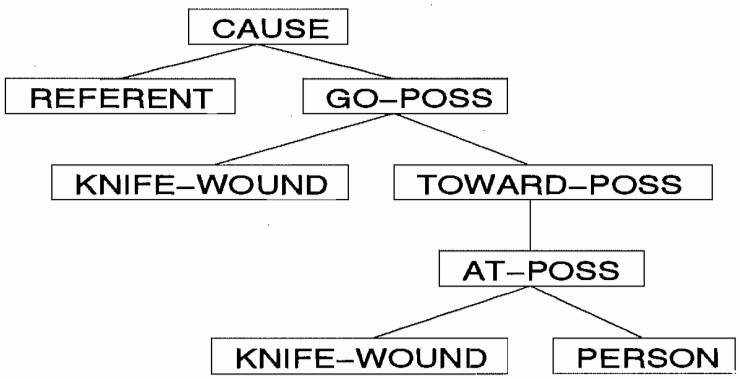


Figure 2: Underlying LCS for Chinese verb HuaShang

This is the interlingua that is used as the pivot from source to target language. The underlying form conveys the meaning that a *referent* cause a *person* to possess a *knife-wound.*

2.3. A Chinese LCS in UNITRAN

Taking UNITRAN as an example. UNITRAN used LCS (Jackendoff, 1991) and Pinker's verb representation with manner (Pinker, 1991) to represent verb semantics. For the Chinese sentence "小明跑步上学 (XiaoMing PaoBu Shang Xue)", the logic form of the whole sentence is derived from the verb semantic representation of "跑步 (PaoBu)". The LCS representation i.e. the argument structure of "跑步 (PaoBu)" is:

```
(DEF-ROOT-WORDS (GO-LOC Y (FROM-LOC (AT-LOC Y Z1)) (TO-LOC (AT-LOC
Y Z2)))
        :ROOTS ((跑步 (Y (* Y))
                (Z1 :OPTIONAL ((* FROM-LOC) (AT-LOC (Y) (Z1))))
                (Z2 (UC (CASE ACC)) ((* TO-LOC) (AT-LOC (Y) (Z2))))
                (MODIFIER JOGGINGLY))
```

This representation defines "跑步 (PaoBu)" falling into the class of GO-LOC. GO-LOC is a three place predicate which represents "motion with manner". The definition of "跑步 (PaoBu)" can be read as "Y is in a motion from location Z1 to a location Z2 with a

'JOGGINGLY' manner". By not arguing on the semantic representation schemes, let's see how the surface structure "小明 跑步 上 学 (XiaoMing PaoBu Shang Xue)" can be analyzed to form a Logic form LCS representation. According to the X-bar theory and the principles like:

I-MAX − > V-MAX N-MAX
V-MAX − > V P-MAX
P-MAX − > P N-MAX

The GB parse tree of the sentence is shown in Figure 3, it is derived by X-bar theory and the other constraints. The composed LCS of the sentence is shown in Figure 4.

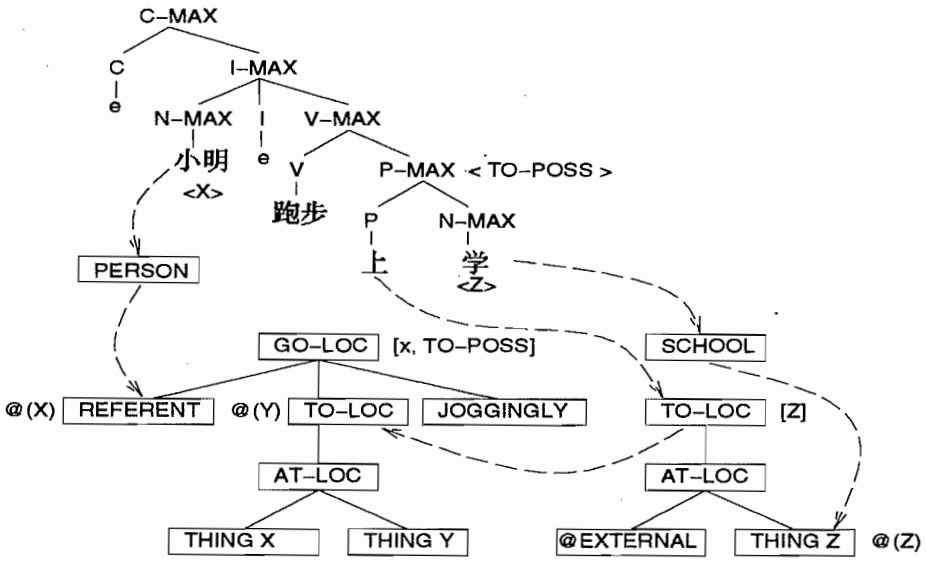
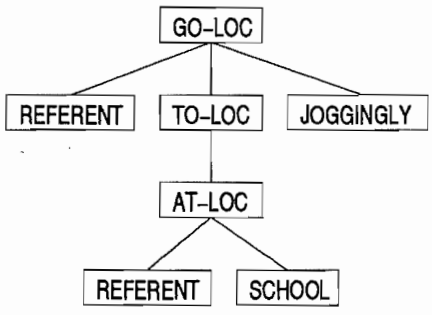


Figure 3: Parse tree and θ role assignment



GO–LOC(REFERENT TO–LOC (AT–LOC (REFERENT SCHOOL)) JOGGINGLY)

Figure 4: Logic form i.e. LCS representation

3. Handling Chinese in UNITRAN

Based on GB and LCS, UNITRAN separates the data and program quite well. Followingly, by presenting an example of how UNITRAN processes Chinese sentence, we will discuss some of the good features of UNITRAN. Obviously, we cannot go into every detail of the system. Interested reader please refer to (Dorr, 1990). The overall design is shown in Figure 5.

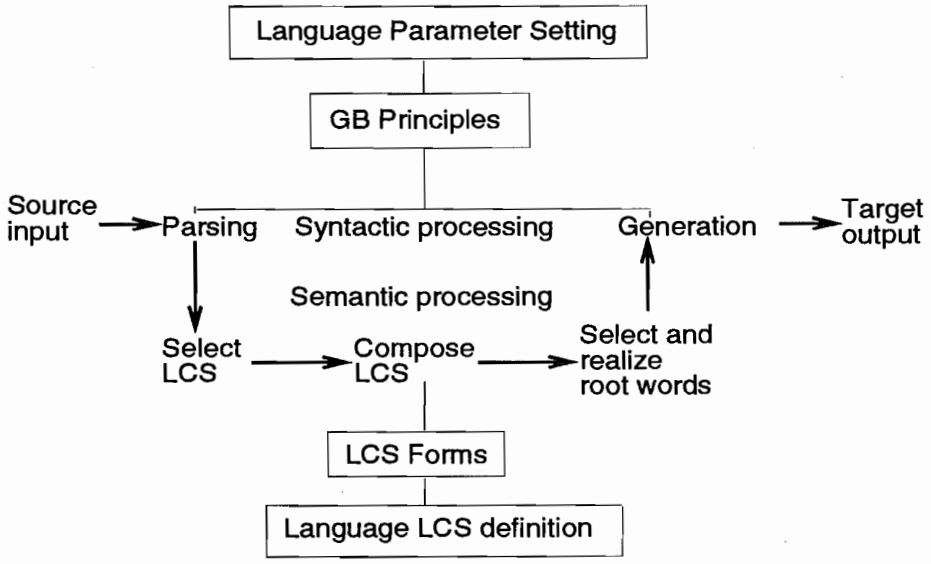


Figure 5: Overall design of UNITRAN

Following is a translation output from English "I stab him" to Chinese.
After Projection and assign X-bar structure, 14 structures left.
Running Translation Examples ...
Parsing (I STAB HIM) ... Done (14 trees in 0.38 seconds).
Assigning X-bar structure to (I STAB HIM) ... Done (14 structures in 0.75 seconds.)
After applying constraints, only one structure left.
Applying Bounding (Trace Linking) ... Done (3 structures in 0.13 seconds.)
Applying X-bar (Feature Matching) ... Done (6 structures in 0.17 seconds.)
Applying Case ... Done (2 structures in 0.25 seconds.)
Applying Binding ... Done (2 structures in 00.20 seconds.)
Applying Theta ... Done (1 structures in 00.20 seconds.)
Following is the parse tree for the sentence.

```
(0
((C-MAX (C "e")
        (I-MAX (N-MAX (N "i")) (I "e")
               (V-MAX (V "stab") (N-MAX (N "him")))))))
```

Lexical conceptual structure is composed.
Composing LCS ... Done (1 structures in 3.18 seconds.)

Following is the LCS structure for the sentence.

(0
(CAUSE REFERENT
(GO-POSS KNIFE-WOUND
(TOWARD-POSS (AT-POSS KNIFE-WOUND REFERENT)))
WITH-INSTR))

Generation begins.

Generating ...

After lexical selection, assign X-bar structure to the generated sentence.

Assigning X-bar structure to 我自己 划伤 他 ...
Assigning X-bar structure to 我的 划伤 他 ...
Assigning X-bar structure to 我 划伤 他 ...
Done (3 structures in 0.50 seconds.)

Apply constraints to the generated structures.

Applying X-bar (Feature Matching) ... Done (6 structures in 00.80 seconds.)
Applying Case ... Done (1 structures in 0.13 seconds.)
Applying Binding ... Done (1 structures in 0.00 seconds.)
Applying Theta ... Done (1 structures in 00.20 seconds.)
Applying Bounding (Trace Linking) ... Done (1 structures in 00.20 seconds.)

Only one legal structures left. Following shows the parse tree for translated Chinese sentence.

(0
((C-MAX (I-MAX (N-MAX (N "我")) (I "e")
(V-MAX (V "划伤") (N-MAX (N "他"))))
(C "e"))))

The feature matching is shown the the following figure:

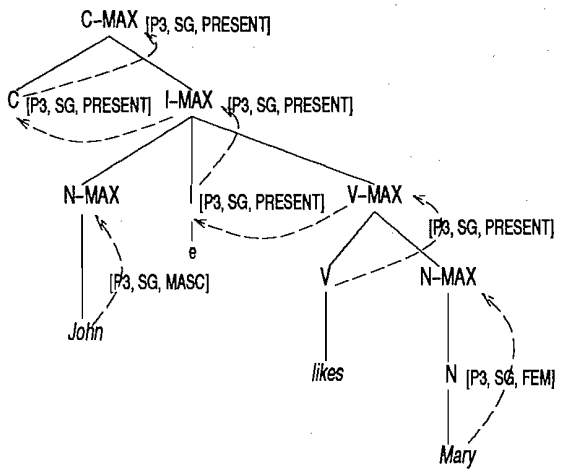


Figure 6: Feature matching in X-bar module

The target language generation is divided into two steps. One is the lexical selection. The

other one is the syntactic realization. This is done by matching the underlying LCS to the appropriate root word from the target language possible set. The lexical selection of Spanish root word for the *stab* event is shown in Figure 7.

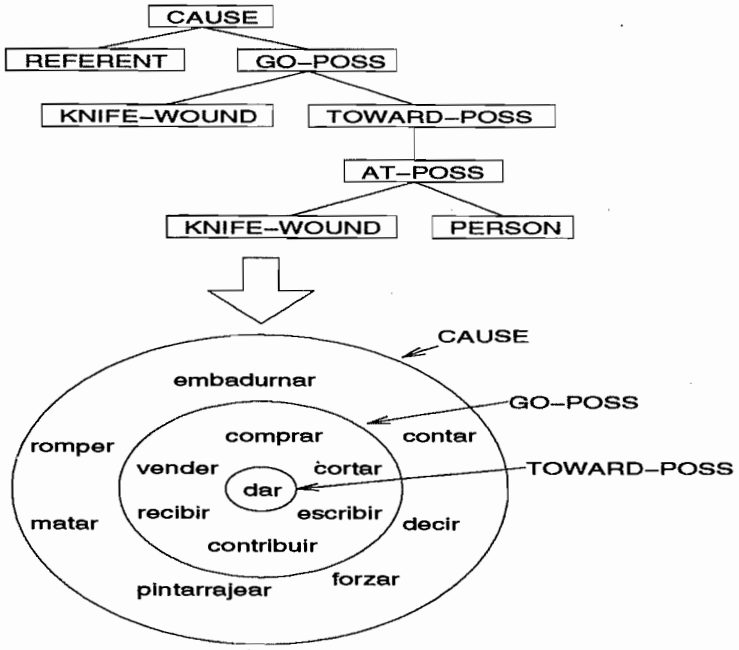


Figure 7: Lexical selection of Spanish root word for English stab event

4. Chinese Parameter Setting

We now come to the main part of the paper: setting those parameter files for Chinese. UNITRAN uses PC-KIMMO as its morphological processor. PC-KIMMO is known as a good tool for morphological processing on languages which have inflections. We let UNITRAN skip the KIMMO processing for Chinese, since Chinese don't have inflections. Since we follow the examples of English verb definition to define Chinese verbs. The main focus will be on Chinese GB parameter setting. In UNITRAN, there is already a set of modules for the GB theory with English, Spanish and German. What we have done is to set those parameters related to Chinese GB theory.

4.1. X-bar module

4.1.1. Basic categories

The choice of X are determined by the basic categories. For Chinese, we employ the view of (Tang, 1989) to classify Chinese language into eight categories. We put Chinese Adverb and Adjective together. Noun phrase is further divided into Determiner phrase and Qualifier phrase. Tense and Aspect is the head for I phrase. The sentence particle is the head for C. The basic categories parameter is set as follows:

Language	Basic categories
Chinese	C, I, V, N, P, A, Q, D

Table 1: Basic categories for Chinese

4.1.2. Constituent Order

The constituent order parameter accounts for the word order distinctions among different languages. According to Tang T. C. , for Chinese, the noun phrase is head final. Adjective phrase is head initial for transitive adjective, head final for intransitive adjective. Proposition phrase is head initial. Complementizer is head final. Determiner is head initial, specifier initial. Qualifier phrase is head initial and specifier initial. We differ with Tang T. C. in viewing that verb phrase is head initial. The constituent order parameter is set as follows:

Category	Chinese
I	SPEC-INITIAL, HEAD-INITIAL
N	SPEC-INITIAL, HEAD-FINAL
C	SPEC-INITIAL, HEAD-FINAL
A	SPEC-INITIAL, HEAD-INITIAL
P	SPEC-INITIAL, HEAD-INITIAL
D	SPEC-INITIAL, HEAD-INITIAL
Q	SPEC-INITIAL, HEAD-INITIAL
V	SPEC-INITIAL, HEAD-INITIAL

Table 2: Constituent order for Chinese

4.1.3. Base-Generated Specifiers

There are two types of specifiers: ones that are base generated in θ position and ones that are moved to a θ-bar position. The base-generated specifiers are assumed to be optional unless the :OBLIGATORY marker is included in the parameter setting. For noun phrase, the specifiers can be a noun phrase or the determiner phrase. For qualifier phrase, the specifiers must be a number. For determiner, the specifier can be the noun phrase. The base-generated specifiers parameter is set as follows:

Category	Chinese
I	N-MAX
N	DET, N-MAX
C	N-MAX
D	N-MAX :OBLIGATORY
Q	NUM :OBLIGATORY

Table 3: Base generated specifier for Chinese

4.1.4. Base-generated Adjuncts

The base adjuncts parameter specifies the position (left, right or free) and the level (minimal or maximal) of each adjunct with respect to the category to which it is adjoined. The base-generated specifiers parameter is set as follows:

Category	Position	Chinese Adjuncts
N	LEFT-MAX	Q-MAX, A-MAX
N	RIGHT-MAX	C-MAX
V	LEFT-MAX	ADV
V	FREE-MAX	P-MAX
A	LEFT-MAX	ADV, P-MAXD
A	RIGHT-MAX	C-MAX
I	LEFT-MAX	ADV, N-MAX, P-MAX
C	LEFT-MAX	ADV, N-MAX, P-MAX

Table 4: Base-generated adjuncts for Chinese

4.1.5. Complements

The Chinese complements parameter is is set as follows:

Category	Chinese Complements
V	(N-MAX), (P-MAX) (P-MAX P-MAX) (N-MAX P-MAX) (C-MAX) (P-MAX C-MAX) (C-MAX P-MAX) (A-MAX) (ADV) (V-MAX)
P N	(N-MAX), (Q-MAX) (P-MAX), (C-MAX) (N-MAX) (D-MAX)
Q	(N-MAX)
D	(N-MAX), (D-MAX)
A	(C-MAX), (N-MAX)
I	(V-MAX), (A-MAX)
C	(I-MAX)

Table 5: Complements for Chinese

4.2. Government Parameter

Government parameter is a key point to those constraints.

4.2.1. Governors

The governors for each Chinese categories are:

Language	Governors
Chinese	V, N, A, P, Q, D, ASP, PAR

Table 6: Governors for Chinese

Here, ASP is for aspect and tense, PAR is for sentence particles.

4.2.2. Proper Governors

The proper governors for Chinese are:

Language	Governors
Chinese	V, P, ASP

Table 7: Proper Governors for Chinese

4.3. Bounding

4.3.1. Bounding node

The bounding node in Chinese are:

language	Bounding nodes
Chinese	I, N

Table 8: Bounding node for Chinese

4.3.2. Moved Specifiers

Category	Chinese Moved Specifiers
I	N-MAX
C	N-MAX, P-MAX

Table 9: Moved Specifiers for Chinese

4.3.3. Moved adjuncts

Category	Position	Chinese Moved Adjuncts
I-MAX	LEFT-MAX	ADV, P-MAX

Table 10: Moved Adjuncts for Chinese

4.4. Trace

The parameters for Trace in Chinese is set as follows:

Trace parameter	Chinese
Traces	N-MAX, P-MAX
Empties	N-MAX in Specifier of I

Table 11: Trace for Chinese

5. Discussion

In the last section, we have shown some of the parameters set for Chinese GB grammar. Several sample Chinese sentences have been successfully run on UNITRAN. This shows that a new language can be easily added into the system just by setting parameters for those principles. However, the merit comes together with the deficits. The requirement of highly abstracted principles for all human languages is very difficult to meet. The Chinese grammar we set in UNITRAN is by no mean a complete one. Although there is some universal rules for human languages to form a core grammar, each particular language has its own idiosyncrasy. These 'periphery' phenomena need the system to handle them piece by piece(Tang, 1989). Unfortunately, the number of irregularities is very larger than the number of principles. Therefore more effort is needed to show that the PBMT style of UNITRAN is suitable to scale up for unrestricted text.

REFERENCES

CHOMSKY, N. (1956). *Syntactic Strcutures.* Mouton.

CHOMSKY, N. (1965). *Aspects of the theory of syntax.* MIT Press.

DORR, B. J. (1990). *Lexical Conceptual Structure and machine Translation.* PhD thesis, MIT.

DOWTY, D. (1991). Thematic Proto-roles and Argument Selection. *Language*, 67(3).

DOWTY, D. R. (1979). *Word Meaning and Montague Grammar.* D. Reidel Publishing Company.

HUANG, J. (1982). *Logic Relations in Chinese and the Theory of Grammar.* PhD thesis, MIT.

HUDSON, R. (1984). *Word Grammar.* Blackwell.

HUTCHINS, W. J. & SOMERS, H. L. (1992). *An Introduction to Machine Translation.* Academic Press, London.

JACKENDOFF, R. (1991). *Semantic Structures.* MIT Press.

LEVIN, B. (1987). Approaches to Lexical Semantic representation. In LEVIN, B., editor, *Readings for Lexical Semantics.* Northwestern University.

LEVIN, B. (1992). English Verb Classes and Alternations: A Preliminary Investigation. Technical report, Department of Linguistics, Northwestern University, 2016 Sheridan Road, Evanston, IL 60208.

LI, Y. H. A. (1990). *Order and Constituency in Mandarin Chinese.* Kluwer Academic Publishers.

NIRENBURG, S., CARBONELL, J., TOMITA, M., & GOODMAN, K. (1992). *Machine Translation: A Knowledge-Based Approach.* Morgan Kaufmann Publishers.

NIRENBURG, S. & NIRENBURG, I. (1988). A Framework for Lexical Selection in Natural Language Generation. In *COLING88.*

PALMER, M. (1990a). Customizing verb definitions for specific semantic domains. *machine Translation*, 5(30).

PALMER, M. & POLGUÈRE, A. (1992). A Computational Perspective on the Lexical Analysis of Break. In *Proceedings of the Workshop on Computational Lexical Semantics*, Touluse, France.

PALMER, M. S. (1990b). *Semantic Processing for Finite Domains.* Cambridge University Press.

PINKER, S. (1991). *Learnability and Cognition.* The MIT Press.

PUSTEJOVSKY, J. (1991). The Generative Lexicon. *Computational Linguistics*, 17(4).

SELLS, P. (1985). *Lectures on Contemporary Syntactic Theories.* CSLI.

SOWA, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley.

TAN, C. L., HSU, L. S., & WU, Z. (1992). On Self-organized approaches to NLP. Technical report, Department of Information System & Computer Science.

TANG, T. (1989). Principle and Parameter Grammar and the comparative analysis between Chinese and Chinese. In *Singapore Symposium on the World Chinese Teaching.*

TANG, T. C. (1992). Grammar theory and Machine Translation: Principle and parameter Grammar. In *ROCLING V R.O.C. Computational Linguistics Conference V.*

TOMITA, M., editor (1991). *Current Issues in Parsing Technology.* Kluwer Academic.

WU, Z., HSU, L. S., & TAN, C. L. (1992). A Survey on Statistical Approaches to Natural Language Processing. Technical Report TRA4/92, Department of information system and computer science, National Universoty of Singapore. Submitted to *computational lingistics.*

WU, Z., PALMER, M., HSU, L. S., & TAN, C. L. (1993). Toward the Similarity-based Fuzzy Lexicon. Technical Report TRE4/93, Department of Information System and Computer Science, National Universoty of Singapore.